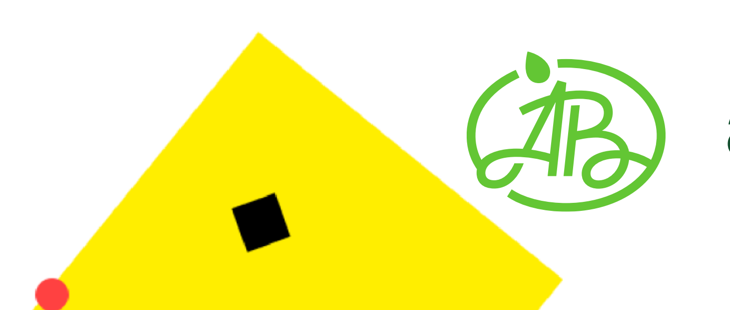
Как мы в Азбуке вкуса микрофронты и UI Kit создавали

Родичкин Данила

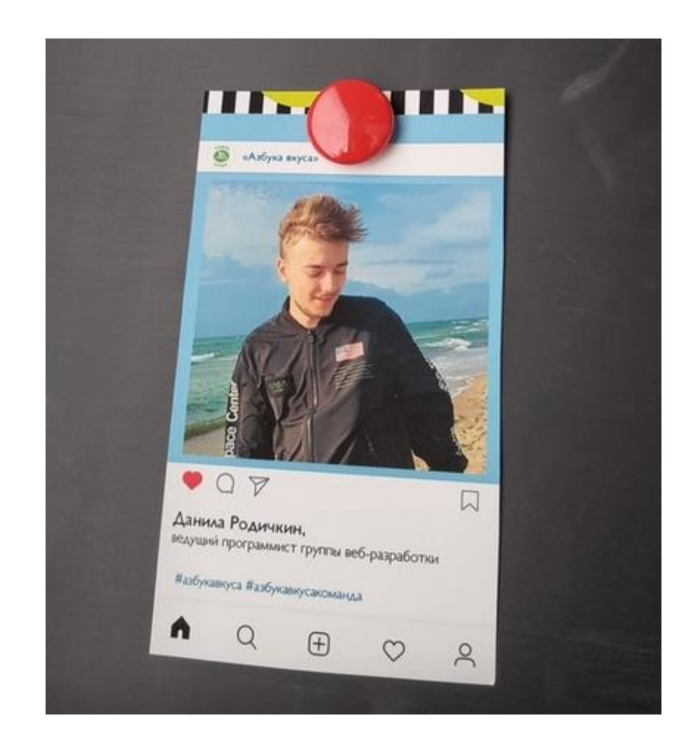




азбука вкуса

Я – Данила Родичкин

- Начинал с фриланса
- Первая работа – рекламное агентство
- В Азбуке вкуса с 2021 года
- Сейчас – руководитель направления фронтенд-разработки







Это – доклад-история про:

- Фронт в Азбуке вкуса
- Две итерации микрофронтов
- UI Kit
- То, как это всё сейчас работает
- То, куда мы идем дальше





Доклад не про:

- Пошаговые инструкции
- Советы из серии «100% варик отвечаю прокатит»





Если тебе интересно узнать:

- Про наш опыт и наши боли
- Про реализацию на нашем стеке (Vue/Nuxt)
- Как выглядит большая система микрофронтов на проде
- Как мы искали способы обмана нашего движка и Webpack
- Из чего состоит наш UI Kit
- Как мы внедряли CSS переменные и SVG компоненты





Кто мы?

«Азбука вкуса» — это уникальная омниканальная экосистема питания с собственными фуд-технологиями и фуд-сервисами

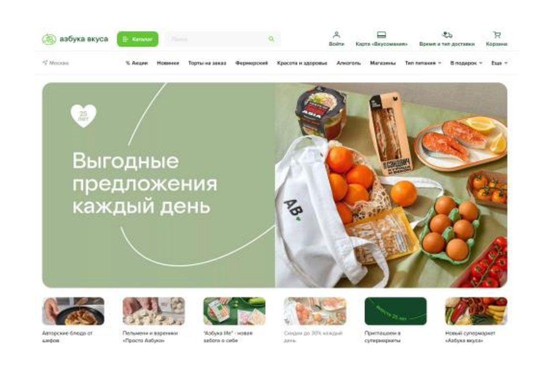
174 магазина

104 супермаркета «Азбука вкуса»,
62 минимаркета «Азбука daily»,
3 специализированных винных «Энотеки»,
5 гипермаркетов «Азбука маркет»,
интернет-магазин av.ru,
городское кафе «АВ Гранд-кафе»

сайт av.ru и приложение «Азбука вкуса»

Уникальный ассортимент: более 23 000 товаров из 70 стран
Удобные способы доставки: широкий ассортимент ко времени, быстрая доставка от 30 минут и самовывоз из магазина









Кто мы внутри?

Системы хранения данных

Siebel – система по работе с программой лояльности	Navision – интегрированная система для управления розничными процессами торговли (ERP, WMS, MES системы)

Kafka

Распределенный брокер для обмена сообщений между системами

Смежные системы

Платежный шлюз	Сервис единой авторизации	Сервис обработки заказов от различных систем

Бэкенд

SAP Hybris, бэкенд для работы сайта и мобильного приложения	Сервисы для обслуживания программы лояльности	Сервисы для спецпредложений клиентам (торты на заказ, продажа эл. карт и т.д.)

Фронтенд

AV.ru	Мобильное приложение	B2C- и B2B-продажи	Приложение для сотрудников





Вступление

В начале был jQuery…





В начале был jQuery…

Экспресс-меню

- Шаблонизатор Twig
- Глубокая, но понятная структура
- Docker
- Чистый JS

Хуже быть не может

Корп. часть (лендинги)

- Шаблоны Bitrix
- Все вперемешку
- Полу-живой Docker
- jQuery
- Два Webpack

Хуже быть не может

AV.ru

- SAP Hybris
- Структура - «без комментариев»
- Docker отсутствует ☹
- Два шаблонизатора






jQuery
Подрядчики
Я

А что не так?

- Всё было разным
- Фронтендеру было необходимо поднимать бэкенд
- «Компоненты» были реализованы через копирование кода

А иногда код забывали скопировать





Выбор стека

- Большая часть команды – подрядчиков – уже работала на Vue
- Vue – мой стек

- Nuxt был выбран как SSR-фреймворк
- На проекте также сделали TypeScript из коробки



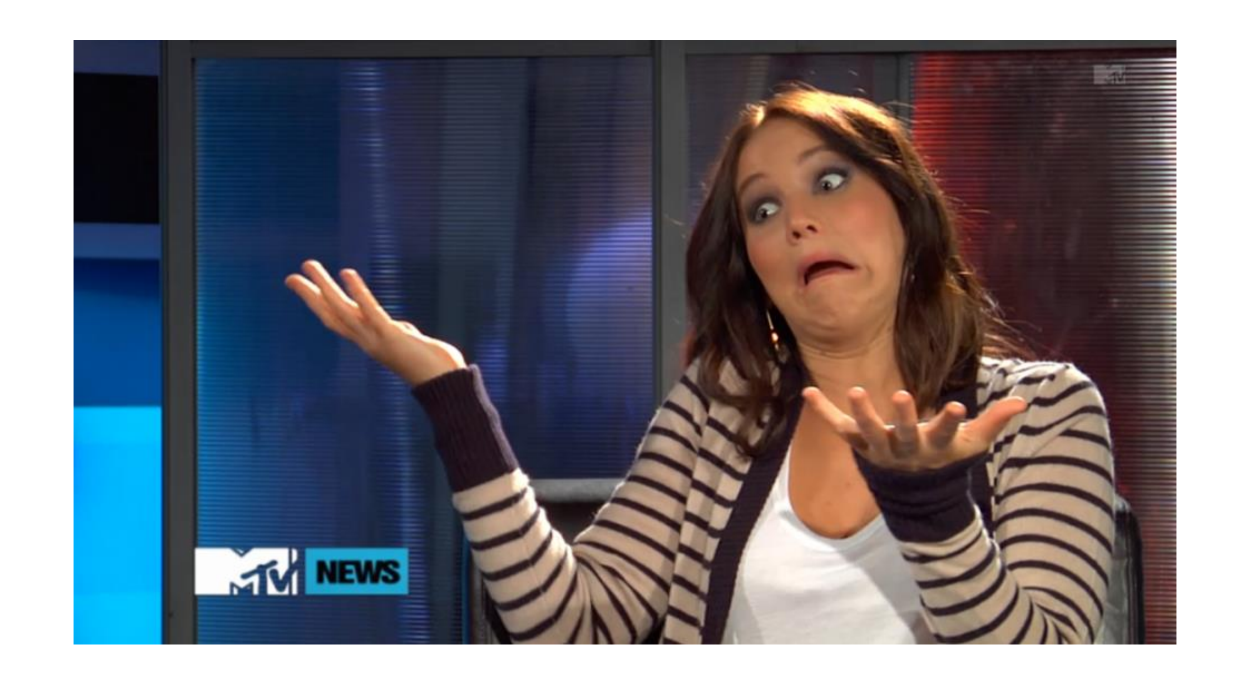






Почему Vue 2?

- Vuetify 3 – нет
- Nuxt 3 – нет
- Pinia – не стандарт по умолчанию
- Vite – сырой
- Vue 3 – не стандарт по умолчанию





Попытка номер раз

Да почему движок сопротивляется?!





Зачем?

- Основной сайт и корп. часть предстояло объединить
- Ускорение сборки
- Разделение структуры
- Поддержка версионирования
- Разделение иконок





С чего начали?

- Один репозиторий – один микрофронт
- Отдельная библиотека для связи воедино фронтов и их настроек
- Каждый фронт собирается своим Webpack
- Компоненты не собираются





Как это работало

- Routes – this.nuxt.extendRoutes – задокументировано
- SCSS – меняем настройки.
  this.nuxt.options.styleResources.scss и this.nuxt.options.css
- Глобальные фукнции – this.nuxt.addPlugin – задокументировано
- Vuex – ох.





Vuex

```
//Шаблонизатором открываем цикл из недокументированной переменной
<% options.keys.split(',').forEach((key) => { %>
    //Чтобы достать ключ, делаем так же
    context.store.registerModule('<%= key %>', {
    //Вызываем незадокументированную функцию, которая возвращает нечто
    //Вызываем require на нечто и получаем наш объект
    ...require('<%= serializeFunction(options.values[key]).replaceAll("'", '') %>'),
    namespaced: true,
    }, { preserveState: context.isClient });
<% }); %>
```





Основной камень преткновения

- Nuxt 2 местами не документирует свои коробочные методы, которые нужны для его расширения
- Решение – изучение исходников движка и (полу)официальных библиотек



Nuxt Community









Другие проблемы / вопросы

- Hot Reload сделали лютейшими костылями
- Как подключать ассеты из фронтов без сборки?
- Откуда брать tsconfig/webpack config? Или дублировать их?
  А если надо менять настройки?
- Откуда брать смежные компоненты типа шапки?
  Выносить в отдельную репу?
- Как нормально писать тесты?
- Откуда брать Nuxt Config?






MOTHER OF GOD
WHAT HAVE I DONE

Почему все же Nuxt?

- Большое сообщество
- Отлаженная работа SSR
- Большие возможности для расширения
- Высокая документированность раздела Nuxt Internals





Вы что, не знали, что так будет?

- Ассеты: планировали использовать подключение по названию библиотеки
- Дублирование конфигов: решалось бы копированием при запуске с учетом заданных настроек
- Смежные компоненты: всё выносим в отдельную репу и подключаем без сборки
- Hot Reload: я был глуп





Ну а чего тогда?

- Слишком много правил
- Неустойчивая структура
- Мы могли дальше идти с этим решением, если бы хотели





Если…

- Вы делаете какое-то свое решение, проблемы будут всегда
- Вы уперлись в проблемы – выпишите их себе
- «Официального» или готового решения для большинства проблем нет
- Подумайте, что вы движетесь не туда





Миссия невыполнима

- Состав миссии: 1 сеньор, 1 мидл, 2 подрядчика
- Начало миссии: июль
- Сроки миссии: конец осени
- Задачи миссии: редизайн и реплатформинг карточки товара и каталога
- Документированность API: 5%
- Шансы на успех: ?%







Попытка номер два

Один UI Kit и шесть месяцев спустя…





Один UI Kit и шесть месяцев спустя…

- Необходимо воскресить идею
- Монорепа стала обкатанной
- Структура по репозиториям не вышла
- Оставляем ту же концепцию в виде монорепы





Настройки проекта

```
export interface IProjects {
  microModuleTest: {
    routes: '2.0/test',
    icons: IAVRUIconMicroModuleTest,
  };
  homePage: {
    routes: '/' | '2.0/demo/sliders',
    icons: IAVRUIconHomePage,
  };
  catalog: {
    routes: 'catalog' | 'search' | 'discount' | 'brands' | 'collections'
    icons: never,
  };
```

- Все фронты выносим в интерфейс
- Для строгости прописываем, какие есть пути и иконки





Настройки проекта

```
const config: IProjectConfig<'singleCart'> = {
   name: 'singleCart',
   package: {
      name: '@av.ru/single-cart',
      version: '2.5.0-dev',
   },
   extendRoutes: (routes, resolve) => {
      routes.push({
         name: 'Cart',
         path: '/cart',
         component: resolve(__dirname,
'pages/index.vue'),
      });
      return routes;
   },
};
```

азбука вкуса



Настройки проекта

```
const config: IProjectConfig<'singleCart'> = {
   scssVariables: [
      {
         src: __dirname + '/scss/singleCartVariables.scss',
         strategy: 'push',
      },
   ],
   routesRegExp: {
      cart: /^\/cart(\/$|$)/,
   },
   vuex: {
      index: indexStore,
      cart: singleCartStore,
   },
   mixins: [
      {
         key: '$singleCart',
         mixin: SingleCartMixin,
         initAndBind: true,
      },
   ],
};
```





Подключение Vuex

- Необходимо регистрировать и дерегистрировать модули «на лету»
- Регистрацию надо проводить в Middleware,
- Дерегистрацию – в routes.afterEach после Vue.nextTick





Это не рекомендуется

- Есть Pinia, new standard tool
- Это не всегда хорошо и предсказуемо работает





NPM-пакеты

- 1 пакет = 1 фронт
- Хранилище пакетов – Gitlab NPM Registry
- Локально и при сборке можно указать, какие пакеты собирать
- При обновлении пакета версия равна версии всей репы





NPM-пакеты

- На тестовых окружениях публикуем по коммиту
- На проде – по фиксированной версии
- На CI собираем дважды





NPM-пакеты

@av.ru/catalog
2.5.0-dev-6e8ec143 published by Данила Родичкин npm
release/2.5 6e8ec143
Created 2 days ago

@av.ru/recent-items
2.5.0-dev-6e8ec143 published by Данила Родичкин npm
release/2.5 6e8ec143
Created 2 days ago

@av.ru/lp-quality
2.5.0-dev-6e8ec143 published by Данила Родичкин npm
release/2.5 6e8ec143
Created 2 days ago

@av.ru/lp-mobile
2.5.0-dev-6e8ec143 published by Данила Родичкин npm
release/2.5 6e8ec143
Created 2 days ago





NPM-пакеты

@av.ru/catalog

v 2.5.0-dev-6e8ec143 published 2 days ago

npm 38.21 KiB release/2.5

Detail Other versions

History

@av.ru/catalog version **2.5.0-dev-6e8ec143** was first created 2 days ago

Created by commit #6e8ec143 on branch **release/2.5**

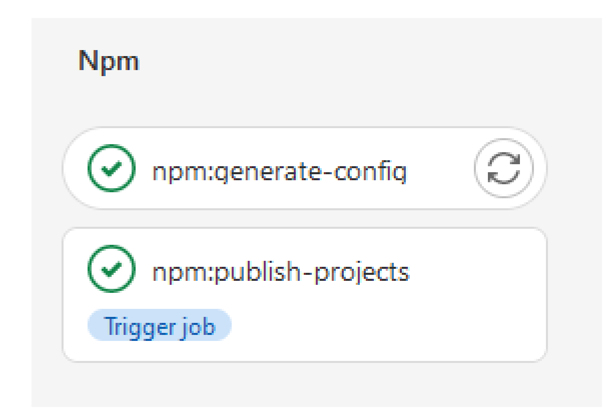
Built by pipeline #93722 triggered 2 days ago by Данила Родичкин

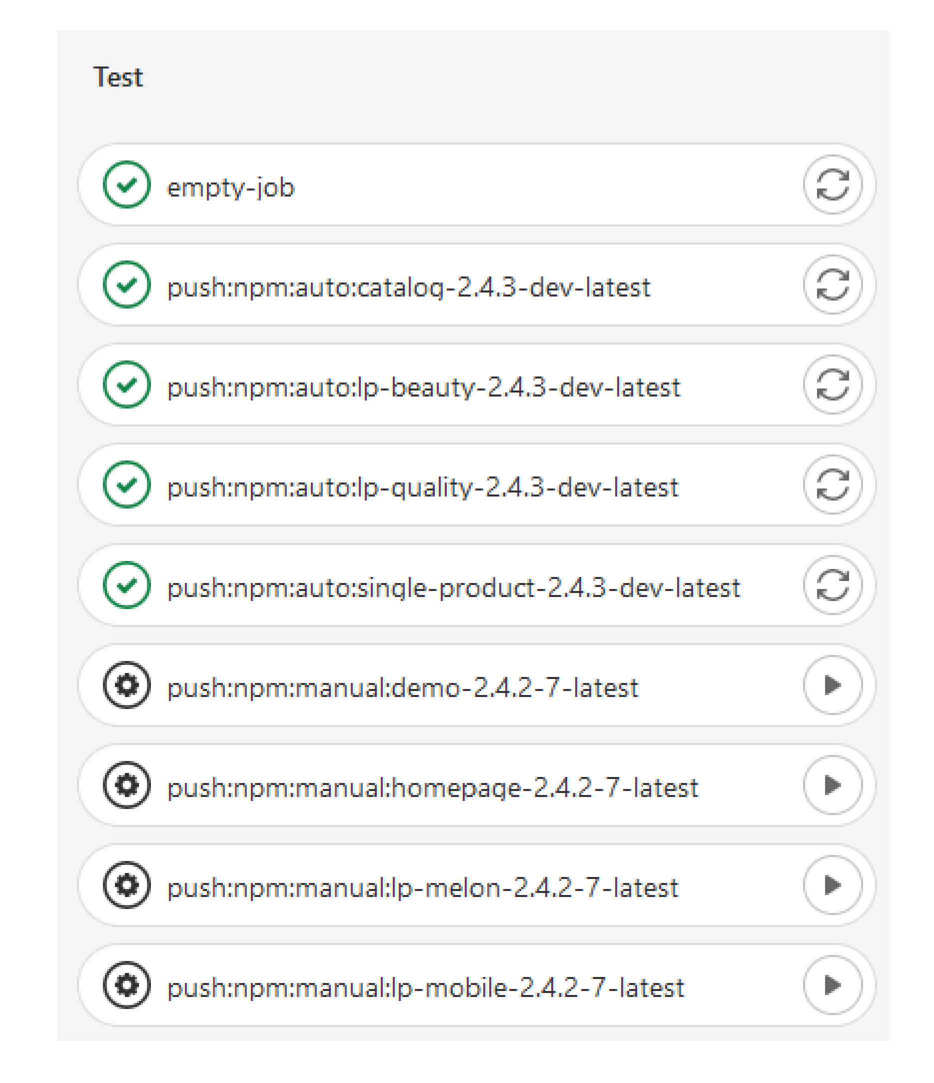
Published to the **Warp** Package Registry 2 days ago





NPM-пакеты


Npm
npm:generate-config
npm:publish-projects
Trigger job


Test
empty-job
push:npm:auto:catalog-2.4.3-dev-latest
push:npm:auto:lp-beauty-2.4.3-dev-latest
push:npm:auto:lp-quality-2.4.3-dev-latest
push:npm:auto:single-product-2.4.3-dev-latest
push:npm:manual:demo-2.4.2-7-latest
push:npm:manual:homepage-2.4.2-7-latest
push:npm:manual:lp-melon-2.4.2-7-latest
push:npm:manual:lp-mobile-2.4.2-7-latest


азбука вкуса


FC
Frontend
Conf 2022

@av/vue-module

UI Kit






ОНИ ДЕЛАЮТ UI KIT.
ЗАЧЕМ?


А! ОНИ БУДУТ УСЛОЖНЯТЬ!
ПРИБАВИТЬ ФИЧ!
РАЗ-ДВА! РАЗ-ДВА!

UI Kit

- Созданные компоненты доступны только на av.ru
- Некоторые компоненты создавались «под задачу»
- Дизайнеры также меняют уже готовый дизайн «под другую задачу»
- Мы изобрели 10 оттенков одного и того же зеленого





Что выносим в библиотеку

- глобальный HTML, подключение шрифтов
- SCSS-переменные
- глобальные классы/методы
- компоненты из Figma UI Kit
- демо-страницу с компонентами
- подключение Swiper, v-touch и других сторонних библиотек





UI Kit

Страницы

Палитра цветов | Кнопки | Инпуты | Всплывающие окна (попапы) | Всплывающие сообщения (тултипы) | Алерты
Virtual Scroller | Infinite Scroll | Лимитированный текст | Иконки | Календарь | Подвал | Типографика | Лоадер

Настройки

XL | primary | Отключена | Разрешить клик по отключенной кнопке | Нет ховера

В состоянии "ховер" | В состоянии "ховер" даже в МВС | Кнопка со ссылкой | noUnderline (для plain) | noFocusAnimation

Вставить иконку слева | Вставить иконку справа

Кнопки

Текст кнопки без иконок | Текст кнопки с иконками | Кнопка с max width, ей мало...

Кнопка с min width/height/кастом padding

Наведись, чтобы текст отодвинулся от стрелки | Кнопка, где изменен основной цвет и цвет текста

Кнопка, где изменены все цвета, включая ховер, но не disabled

Controls





Подключение компонентов

- Для локальной разработки копируем компоненты в папку .global-components
- Добавляем “alias” в Nuxt Config и “paths” в TSConfig
- Компоненты и статика в vue-module не собирается





Работа с цветами

- Основной минус CSS Variables – отсутствие поддержки прозрачности
- А также невозможность доступа к переменным на SSR (из коробки)





Работа с цветами

```
@function toRawRGB($color) {
  @return unquote(red($color) + ', ' + green($color) + ', ' + blue($color));
}

//Основные цвета бренда
$brandPrimaryRaw: toRawRGB(#63C634); //99, 198, 52 на выходе

$colorsMap: (
  //primary
    ibrandPrimary: var(--brandPrimary, $brandPrimaryRaw),
);

$brandPrimary: rgb(map-get($colorsMap, 'ibrandPrimary’));

@function varToRgba($colorName, $opacity) {
  @if (map-has-key($colorsMap, 'i#{$colorName}')) {
    @return rgba(map-get($colorsMap, 'i#{$colorName}'), $opacity)
  } @else {
    @return rgba(var(--#{$colorName}), $opacity)
  }
}
```





Работа с цветами

```scss
@import 'styles';

:export {
  brandPrimary: $brandPrimary;
  brandPrimaryRaw: $brandPrimaryRaw
}
```

```js
import variables from '../../../public/scss/js-variables.scss';
```

```js
{
  test: /\.scss$/,
  use: [
    {
      loader: 'css-loader',
      options: {
        importLoaders: 1,
        modules: {
          mode: 'icss',
        },
      },
    },
    {
      loader: 'sass-loader',
    },
  ],
},
```





brandPrimary:	99, 198, 52
brandSecondary:	18, 79, 46
darkGrey:	35, 37, 39
mediumGrey:	130, 136, 142
lightGrey:	193, 198, 204
superLightGrey:	231, 233, 236
ultraLightGrey:	246, 247, 248
mint:	62, 180, 137
herb:	70, 157, 105
pistachio:	210, 217, 127
cherry:	164, 42, 75
purple:	164, 104, 204
plum:	85, 26, 99
blueberry:	42, 51, 95
milk:	244, 247, 239
matcha:	164, 197, 178
lagune:	106, 186, 185
cacke:	184, 149, 122
peach:	245, 148, 90
cranberry:	223, 83, 83
orchid:	181, 156, 198
royal:	39, 83, 105
marine:	84, 160, 194
oliva:	199, 214, 175
sky:	114, 196, 220
cream:	241, 191, 156
melon:	249, 203, 58
rose:	232, 156, 169
lavender:	222, 207, 230
blue:	146, 196, 231
moon:	233, 245, 231
tiffany:	175, 226, 222
cappuccino:	254, 234, 212
vanilla:	247, 221, 158
sakura:	251, 202, 202
pearl:	237, 238, 249
ice:	237, 246, 255
yellow:	253, 214, 75
red:	198, 52, 52
white:	255, 255, 255
black:	0, 0, 0
containerDesktopWidth:	1200px
containerMobilePadding:	20px
pageWidth:	100vw
pageWindowWidth:	100vw
pageHeight:	100vh
pageWindowHeight:	100vh
pageScrollbarWidth:	0px

От SVG хочется, чтобы:

1. цвета SVG автоматически из HEX становились CSS Variables
2. SVG можно было импортировать как компонент
3. каждый цвет SVG можно было менять параметром компонента
4. набор SVG был разным у каждого фронта
5. цвет SVG можно было менять через изменение color (currentColor)





Проблема стандартных библиотек

- нужна возможность делить на микрофронты
- нужна интеграция с палитрой цветов





Code splitting для SVG

- глобальная переменная с набором иконок (набор = name проекта)
- в index.vue вызывается миксин, куда передается require-функция





Code splitting для SVG

```
mixins: [
  createProjectIconsMixin({
    project: 'homePage',
    requireFunction: (icon: string) => require(`./../assets/icons/${ icon }.svg?prefix`),
  }),
],
```

```
export function createProjectIconsMixin({
  project,
  requireFunction,
}: {
  project: IProjectsList,
  requireFunction: (icon: string) => any,
}): ComponentOptions<Vue> {
  return {
    beforeCreate() {
      if (!commonIconsList[project]) {
        commonIconsList[project] = (icon: string) => {
          return requireFunction(icon.replace(`${ project }/`, ''));
        };
      }
    },
  };
}
```





Алгоритм svgMultipleColors

```
attributes.class = `__transformed-icon--advanced_${ name }-${ index }`;
if (!attributes.style) attributes.style = '';

if (avColor) {
   attributes.class += ` __transformed-icon--advanced_${ name }-${ avColor }`;
   attributes.style += `${ name }: var(--icon-color-${ avColor }, var(--icon-color-${ index }, ${ AVPlatformStyles.variables[avColor] }));`;
}
else {
   attributes.style += `${ name }: var(--icon-color-${ index }, ${ attribute });`;
}
```

- Иконка преобразовывается библиотекой svgo:
  добавляется viewBox вместо width,height и прочие оптимизации
- Иконка парсится в js через библиотеку svgson
- Рекурсивно проходимся по всем «детям»





Финал

Подводим итоги





Основные этапы

1-я итерация микрофронтов

UI Kit

Каталог и карточка товара

2-я итерация микрофронтов

Главная и корзина

FrontendConf 2022





Что дальше

Зима 2022

- Axios -> Fetch
- Vuex -> Pinia
- Vue 2.7
- Composables вместо глобальных функций

Весна 2023

- Редизайн страницы оформления заказа
- Постепенный редизайн личного кабинета
- Расширение UI Kit

До конца 2023

- Переход на Vue 3 и Nuxt 3
- (Возможно) Переход на Vite
- (Возможно) Повторное посещение FrontendConf с рассказом о косяках Nuxt 3 и Vite





25 лет «Азбуке вкуса»!

Скидка 25% по промокоду **25LET**





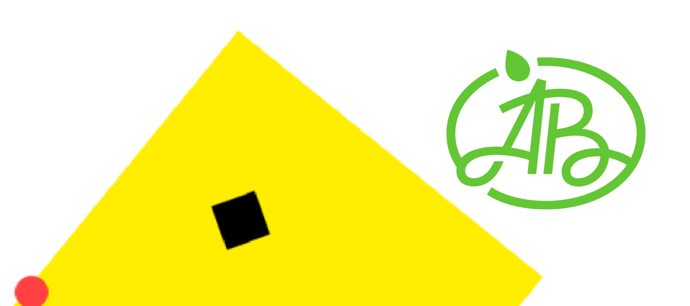


Спасибо за внимание!









азбука вкуса